ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

"COMPARACIÓN DE DOS MÉTODOS DE OBTENCIÓN DE REGLAS DIFUSAS A PARTIR DE CONJUNTOS DE DATOS"

Mikel Ayape López

Tutor: José Antonio Sanz Delgado

Pamplona, 28 de Julio de 2010

ÍNDICE

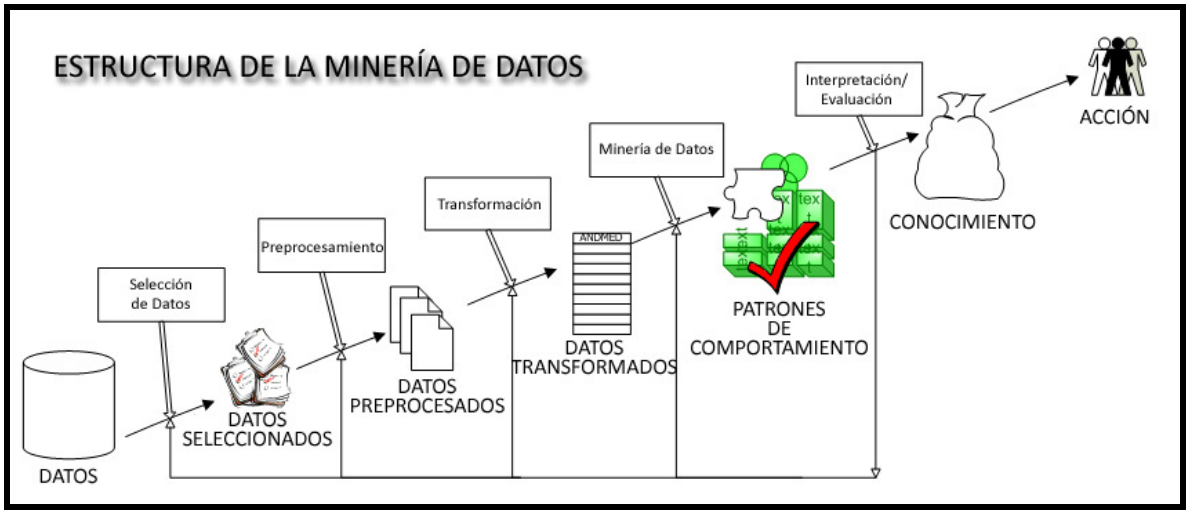
1 - INTRODUCCIÓN

La **minería de datos** es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

Básicamente, la minería de datos surge para intentar ayudar a comprender el contenido de un repositorio de datos. Con este fin, hace uso de prácticas estadísticas y, en algunos casos, de algoritmos de búsqueda próximos a la Inteligencia Artificial y a las redes neuronales.

De forma general, los datos son la materia prima bruta. En el momento que el usuario les atribuye algún significado especial pasan a convertirse en información. Cuando los especialistas elaboran o encuentran un modelo, haciendo que la interpretación que surge entre la información y ese modelo represente un valor agregado, entonces nos referimos al conocimiento.

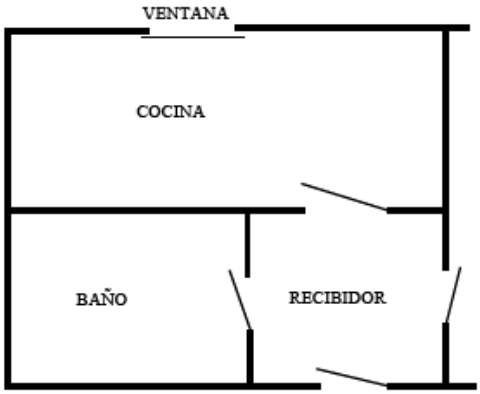
La minería de datos se utiliza en varios campos como pueden ser los negocios (patrones de fuga, fraudes, recursos humanos…), el comportamiento de los usuarios de Internet, para el desarrollo de juegos (ajedrez, tres en raya…), ciencia e ingeniería (genética, ingeniería eléctrica, análisis de gases…), etc.



En este proyecto se aplicarán las técnicas de minería de datos para la extracción de información de un conjunto de datos (pacientes en el primer caso y píxeles de la imagen en el segundo caso), se recibirán una serie de datos en los cuales se buscarán unos patrones de comportamiento para que posteriormente se saquen una serie de reglas difusas que permitirán clasificar el comportamiento de posteriores datos.

Las reglas son el formalismo más común de representar el conocimiento en un Sistema Basado en Conocimiento (SBC). Son sentencias del tipo "Si … entonces …", también conocidas como reglas de producción. Pueden ser interpretadas como "Si *condición P* entonces *conclusión C*", si se cumple el antecedente P se deduce la conclusión C. No obstante, el conocimiento no tiene porque ser categórico y podemos

crear reglas que muestren esa incertidumbre, serán reglas del tipo “Si … entonces … con probabilidad …”. Supongamos el siguiente plano y estamos generando un sistema basado en reglas para detectar fallos en la vivienda.

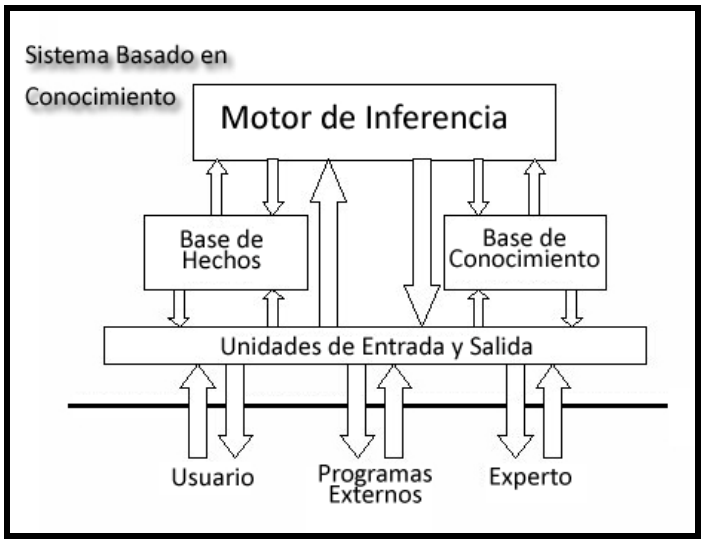


Se pueden crear reglas para nuestro sistema del tipo categórico (regla 1) o no categórico (regla 2).

Regla 1 - **Si** la cocina está seca y el recibidor está mojado **entonces** la fuga de agua está en el baño.

Regla 2 - **Si** el recibidor está mojado y el baño está seco **entonces** el problema está en la cocina **con probabilidad** 0.9

A continuación se muestra el esquema de un Sistema Basado en Conocimiento (SBC) en el cual se diferencian tres partes del sistema, la Base de Hechos (contiene los hechos, por ejemplo: “el recibidor está mojado”), la Base de Conocimiento (contiene las reglas del sistema) y el Motor de Inferencia, que Modela el proceso de razonamiento humano, es decir, aplica las reglas a los hechos existentes en la base de hechos



2 - EXTRACCIÓN DE REGLAS

2.1 - Introducción

En este apartado vamos a estudiar el artículo “An Interpretable Fuzzy Rule-Based Classification Methodology For Medical Diagnosis”, un artículo de Ioannis Gadaras y Ludmil Mikhailov, y posteriormente estudiaremos las posibilidades de aplicar el método descrito en el artículo a la clasificación en imágenes

Los autores escribieron el artículo, según se comenta en el mismo, porque, a pesar de la gran cantidad de implementaciones de sistemas informáticos para la medicina, y a pesar del desarrollo tecnológico seguía siendo difícil la clasificación de los pacientes en función de si éstos estaban enfermos o no. Según el artículo, la dificultad radica en que no hay suficiente conocimiento sobre los mecanismos biológicos y sus interacciones, además de que los resultados médicos y las medidas de los mismos son muy ambiguos. Todo esto hace que la clasificación sea muy complicada.

He elegido este artículo para estudiarlo porque trata el tema de la clasificación de una forma un tanto especial con la creación de hipercubos para clasificar los datos. Además, parece un artículo bastante bueno según los resultados que el autor nos indica que se obtienen siguiendo el método descrito en el mismo.

Posteriormente se va a intentar aplicar el método a un caso distinto, se va a sacar el método del campo de la medicina, para el cual esta diseñado, para aplicarlo a la clasificación en imágenes. Nos encontraremos con diversas dificultades, como que los hipercubos pierden su sentido al encontrarnos trabajando con una única variable, por lo que trabajaremos en un espacio unidimensional, en el cual representaremos mediante intervalos en lugar de hipercubos. También surgirá el problema de que el artículo original está diseñado para obtener tan solo 2 posibles salidas “paciente enfermo” o “paciente sano” y en este caso podremos clasificar un píxel en varias clases distintas, por lo que tendremos que aplicar algún método que nos soluciones ese aspecto.

2.2 - ESTUDIO DEL ARTÍCULO "AN INTERPRETABLE FUZZY RULE-BASED CLASSIFICATION METHODOLOGY FOR MEDICAL DIAGNOSIS"

2.2.1 - Descripción del método

2.2.1.1 – OBTENCIÓN DE CUBOS Y CÁLCULO DE SUS SALIDAS

El método descrito en el artículo *'An interpretable fuzzy rule-based classification methodology for medical diagnosis'* sigue el esquema marcado en la Fig. 1 para la obtención de cubos y cálculo de las salidas asociadas. El esquema tiene 5 procesos diferenciados los cuales iremos explicando a continuación, la existencia de 2 umbrales definidos por el usuario nos irán desplazando entre esos procesos hasta llegar al final de la ejecución del método.

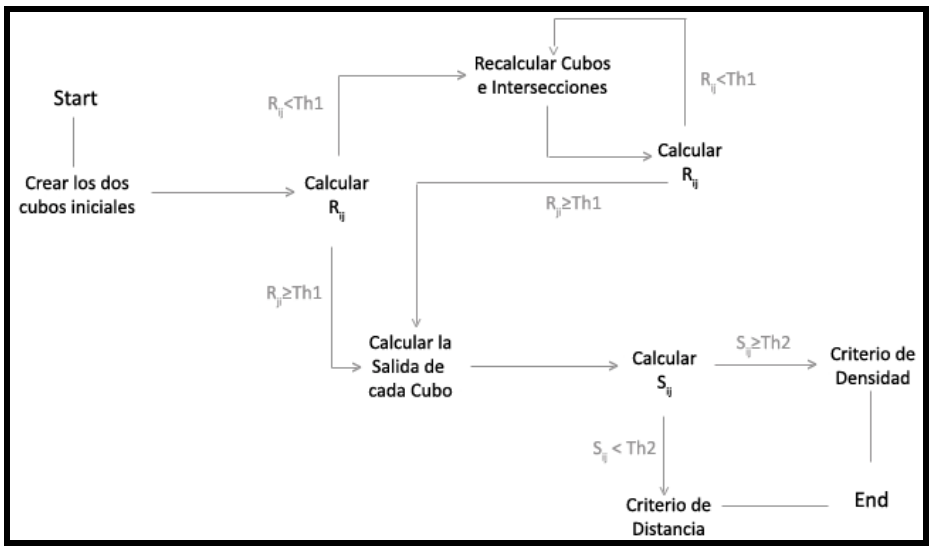


Fig. 1

2.2.1.1.1 – DATOS DE ENTRADA

El método implementado recibirá como entrada un conjunto de datos que representarán los valores de distintos pacientes en aspectos relacionados con la enfermedad a estudiar, así como la salida asociada al paciente, es decir, si está enfermo o no de dicha enfermedad. Por lo tanto, cada dato recibido será de la forma [(X_1, X_2…, X_n, Y)]. Siendo las X_i los valores que tiene el paciente en cada aspecto estudiado y la Y representará la salida.

Recibiremos un conjunto de datos semejante al de la Fig. 2, en el cual, cada conjunto de coordenadas, en este caso tan solo dos para facilitar la visión, representa los datos de un paciente. Si coloreamos los datos de dichos pacientes en función de su salida asociada obtendremos la Fig. 3, en la cual los pacientes enfermos están representados de color rojo y los pacientes sanos están representados en color azul.

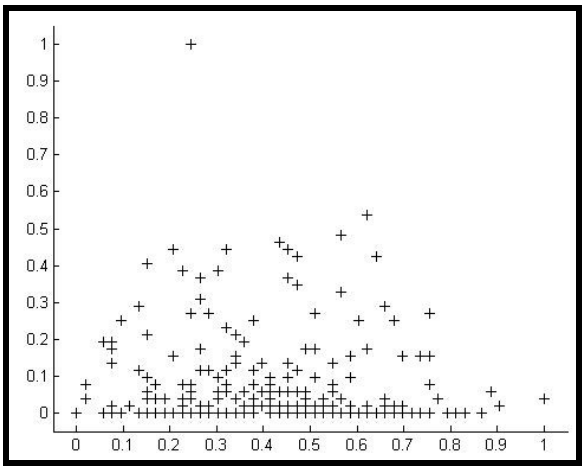


Fig. 2

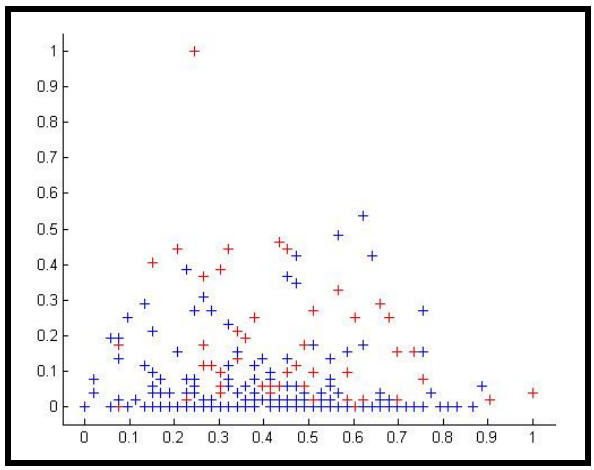


Fig. 3

2.2.1.1.2 – CREACIÓN DE LOS CUBOS INICIALES

Con estos datos vamos a construir hipercubos que nos los vayan agrupando en función de la salida asociada que tengan. Estos hipercubos, inevitablemente, contendrán datos que no pertenezcan a la salida asociada a los mismos. Por lo que, a su vez, serán tratados para mejorar la salida.

El primer paso que realizaremos será la creación de los cubos iniciales, cada cubo inicial deberá englobar todos los datos que pertenezcan a la misma salida, y ese cubo será asociado a dicha salida. En la Fig. 4 vemos como se crearían los cubos para los datos de las Fig. 2 y Fig. 3.

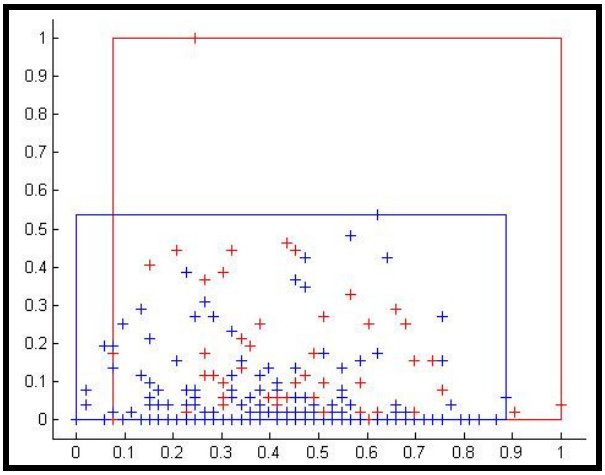


Fig. 4

2.2.1.1.3 – CÁLCULO DEL UMBRAL R_{ij}

Una vez tenemos calculados los cubos iniciales, procederemos a calcular el Umbral R_{ij}, este umbral, nos determinará si se deben obtener nuevos cubos a través de subdivisiones de dichos cubos o si los cubos son suficientemente pequeños. Un valor de R_{ij} próximo a cero indica que la cantidad de datos es muy pequeña con respecto al total

de datos, por lo que no es necesaria una nueva subdivisión del cubo, si el umbral es mayor, contendrá más datos.

El umbral Th1 representará un valor que determinará si se debe seguir subdividiendo dicho cubo, si el umbral R_{ij} del cubo es menor que el umbral Th1, no se subdividirá más, si es mayor, realizaremos una nueva iteración para dividir dicho cubo.

Para el cálculo del umbral R_{ij} aplicaremos la siguiente fórmula:

$$R_{ij} = \frac{D(A_i^{l} \cap A_j^{l})}{D(A_i^{1} \cup A_j^{1})}$$ donde D(x) denota el número de datos pertenecientes a x.

2.2.1.1.4 – RECALCULAR CUBOS E INTERSECCIONES

En este apartado hay dos funciones claramente diferenciadas, mediante ellas obtendremos nuevos cubos.

La primera de las dos funciones es la de intersección. Cuando dos cubos intersecionen entre sí, se creará un nuevo cubo formado por la intersección de ambos.

La segunda de las funciones de este paso se produce en el caso de que uno de los cubos no tenga ninguna intersección para obtener nuevos cubos, por lo tanto, lo que se realiza es volver a dividir dicho cubo en nuevos cubos, definidos por los límites mínimo y máximo de los datos que contiene dicho cubo, de la misma forma que hemos hecho al inicio del método para calcular los dos primeros cubos, pero teniendo en cuenta tan solo los datos pertenecientes al cubo que queremos dividir.

2.2.1.1.5 – CALCULAR LA SALIDA DE CADA CUBO

Una vez llegados a este punto, ya tenemos calculados todos los cubos que surgen de los datos recibidos, tenemos los límites de cada uno de ellos, ahora debemos calcular la salida asociada a cada cubo.

Existen dos criterios para calcular la salida asociada a un cubo, que son el criterio de densidad y el criterio de distancia, dependiendo de los datos que pertenezcan al cubo que queremos estudiar. El criterio de densidad se basará en que el cubo adopte la salida más repetida dentro de los datos que se encuentren entre sus límites. El criterio de distancia se basará en asociar la salida en función de donde se situé el dato, ya que supone que los datos con una salida están agrupados en una zona del cubo.

2.2.1.1.5.1 – CÁLCULO DEL UMBRAL S_{ij}

Una vez tenemos calculados todos los cubos, tendremos que calcular la salida asociada a cada uno de ellos, existen dos métodos para el cálculo de la salida, por ello surge el umbral S_{ij}, su función es determinar cual de los dos criterios es el que se debe utilizar para calcular la salida asociada al cubo.

Este umbral S_{ij} tomará valores entre 0 (cuando el cubo contiene el mismo número de datos de una clase que de la otra) y 1 (cuando el cubo no contiene datos mas que de una de las dos clases). Si el valor de S_{ij} esta próximo a cero, se realizará el cálculo de la salida mediante criterio de distancia, si el valor de S_{ij} está próximo a uno, se realizará el cálculo de la salida mediante criterio de densidad.

El término próximo es ambiguo, por lo que surge el umbral Th2, el cual determinará el límite que indicará si el umbral S_{ij} está próximo a uno o a cero, es decir, determinará si se debe utilizar el criterio de densidad o el criterio de distancia para el cálculo de la salida asociada al cubo.

Para el cálculo del umbral S_{ij} aplicaremos la siguiente fórmula:

$$S_{ij}^{I} = \frac{|D_i(A_i^{I} \cap A_j^{I}) - D_j(A_i^{I} \cap A_j^{I})|}{D(A_i^{I} \cap A_j^{I})}$$

Donde D (x) denota el número de datos pertenecientes a x y Di(x) denota el número de datos de la clase i pertenecientes a x.

2.2.1.1.5.2 – CRITERIO DE DISTANCIA

El criterio de distancia da lugar a un cubo sin una salida definida de antemano, la salida dependerá de la posición del dato dentro del cubo. Si utilizamos este criterio es porque los datos pertenecientes a este cubo están equilibrados, hay, aproximadamente los mismos de cada clase. Lo que hacemos es calcular dos centroides, que representen el punto medio de los datos de cada clase dentro del cubo.

Estos centroides se calculan siguiendo la siguiente fórmula:

$$C(k) = \frac{x_1(k) + x_2(k) + \ldots + x_N(k)}{N}$$ Donde $x_i(k)$ es la coordenada k del dato x_i

En la Fig. 5 vemos como, siguiendo con el ejemplo inicial, al calcular la salida del cubo formado por la intersección de los dos iniciales, tenemos que aplicar criterio de distancia, por lo tanto obtenemos el cubo marcado en magenta, con los centroides

marcados, en rojo el centroide de los datos enfermos (tienen de salida asociada uno) y en azul el centroide de los datos sanos (salida asociada es cero).

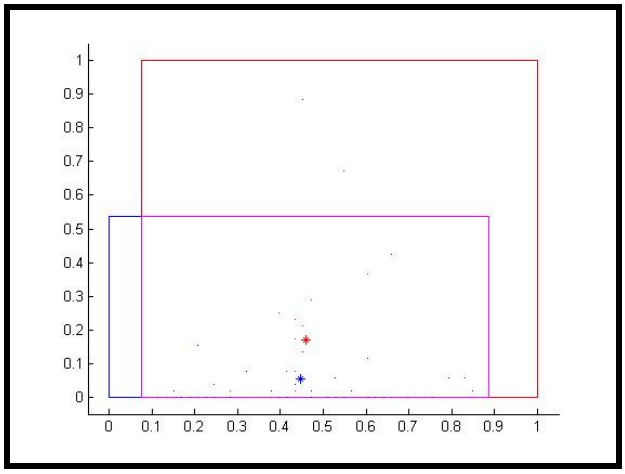


Fig. 5

Para el cálculo de las distancias de un dato cualquiera a los centroides utilizaremos distancias euclídeas, es decir, aplicaremos las siguientes fórmulas:

$$d_i = \sqrt{\left(x_1 - x_1^{Ci}\right)^2 + \left(x_2 - x_2^{Ci}\right)^2 + \ldots + \left(x_m - x_m^{Ci}\right)^2}$$

$$d_j = \sqrt{\left(x_1 - x_1^{Cj}\right)^2 + \left(x_2 - x_2^{Cj}\right)^2 + \ldots + \left(x_m - x_m^{Cj}\right)^2}$$

Las reglas obtenidas con este método serán del tipo:

"IF x is in A_{ij} THEN y is in Y_i when $d_j > d_i$ OR y is in Y_j when $d_i > d_j$"

2.2.1.1.5.3 – CRITERIO DE DENSIDAD

El criterio de densidad asocia una salida fija a un cubo, sea uno o cero, este criterio se aplica en el caso en que al calcular el umbral S_{ij} consideremos que el número de valores de una clase dentro de un cubo es significativamente mayor que el número de valores de la otra clase (determinado mediante el umbral Th2).

Para calcular cual de las salidas es la que debemos asociar al cubo se calculan los dos valores siguientes

$$W_i = \frac{D_i(A_{ij}^l)}{D(A_{ij}^l)} \qquad W_j = \frac{D_j(A_{ij}^l)}{D(A_{ij}^l)}$$

Donde D(x) es el número de datos pertenecientes al cubo x y $D_i(x)$ son el numero de datos de la clase i pertenecientes al cubo x.

A continuación realiza una comparación, si $W_i > W_j$ asocia la salida Y_i al cubo, si por el contrario $W_i > W_j$, asociará la salida Y_j al cubo.

Las reglas obtenidas con este método son del tipo:

"IF x is in A_{ij} THEN y is in Y_i WHEN $W_i > W_j$, OR y is in Y_j WHEN $W_i < W_j$"

Al ser W_i y W_j dos valores fijos una vez los hayamos calculado, podremos sustituir esta regla por una de las dos siguientes.

"IF X is in A_{ij} THEN y is in Y_i" (si $W_j < W_i$)

"IF x is in A_{ij} THEN y is in Y_j" (si $W_i < W_j$)

2.2.1.2 – PROCESO DE INFERENCIA DIFUSA

Una vez llegados a este punto, tenemos calculados los límites del cubo, en este punto es cuando debemos fuzzificarlos, es decir, partiendo del cubo que tenemos calculado debemos "ampliar" sus fronteras, aceptando una cierta pertenencia de cada dato a dicho cubo en función de lo próximo que se encuentre a él. En la Fig. 6 se observa un cubo al cual se le a aplicado este proceso.

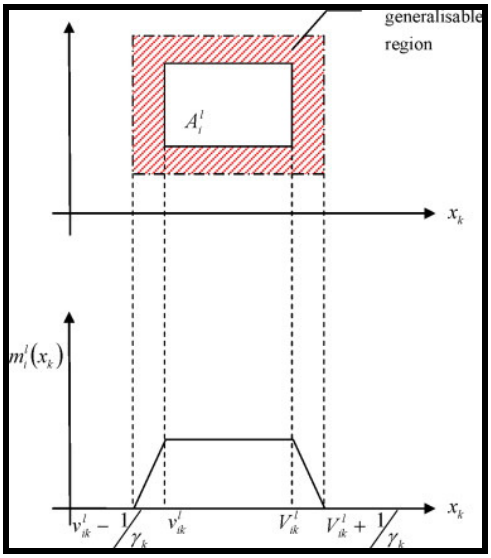


Fig.6

Para calcular la pertenencia de un punto al cubo aplicaremos la siguiente fórmula, que dependerá de un factor configurable por el usuario llamado γ y que nos determinará el tamaño de la zona de pertenencia entre 0 y 1. Es decir, a mayor γ, menor será el tamaño de la zona difusa.

Diferenciaremos entre el caso en que el cubo que vamos a tratar no tiene intersecciones con otros cubos y el caso en que el cubo tiene intersecciones con otros cubos, para calcular la zona difusa.

En el caso en que no hay intersecciones con otros cubos aplicaremos la siguiente fórmula:

$$m_i^l(x_k) = \min\{[1 - \max(0, \min(1, \gamma(v_{ik}^l - x_k)))],$$
$$[1 - \max(0, \min(1, \gamma(x_k - V_{ik}^l)))]\}$$

En el caso con varios cubos, el calculo de la zona difusa sigue otro estilo diferente en la zona en que interseccionan ambos, se realizará como indica la Fig. 7

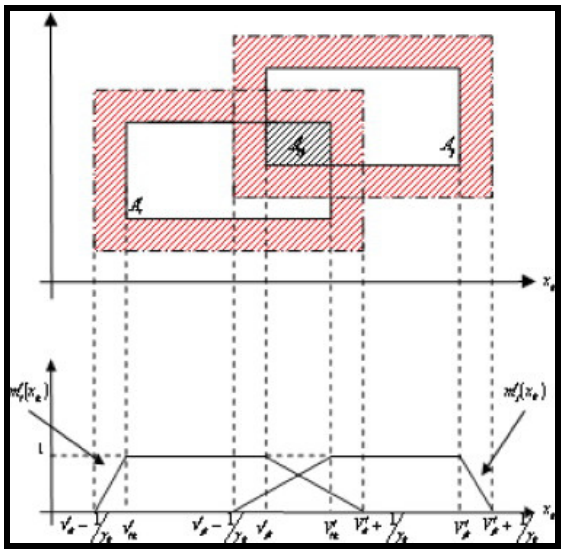


Fig. 7

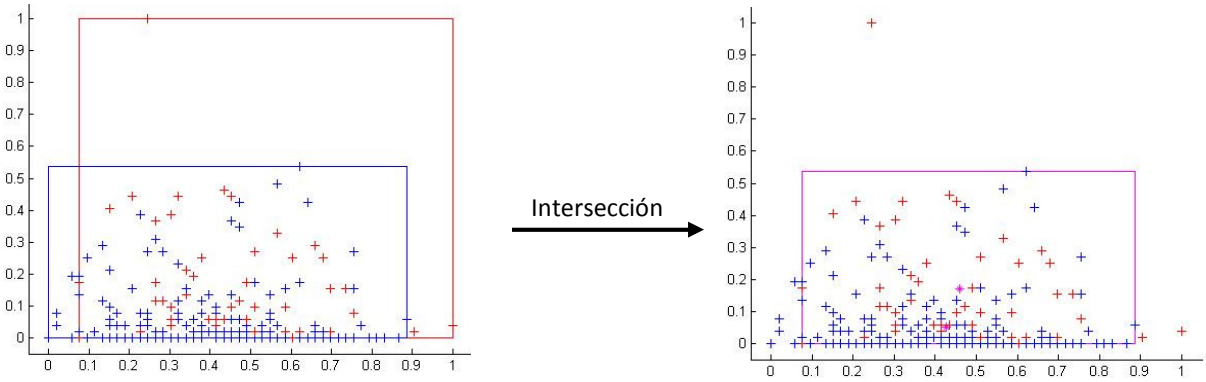
Para calcular estas zonas aplicaremos la siguiente formula, que tiene en cuenta los dos cubos para el cálculo de la zona difusa:

$$m_i^l(x_k) = \min\{[1 - \max(0, \min(1, \gamma(v_{ik}^l - x_k)))],$$
$$[1 - \max(0, \min(1, (1/(V_{ik}^l + 1/\gamma - v_{jk}^l))(x_k - V_{ik}^l)))]\}$$

Cuando tenemos ya realizado este paso tan solo deberemos calcular para cada dato el valor de pertenencia a cada cubo y asociaremos como salida del dato el valor correspondiente a la salida del cubo en el que tenga mayor pertenencia.

2.2.2 - Secuencia de Obtención de los Hipercubos

Cubos Iniciales

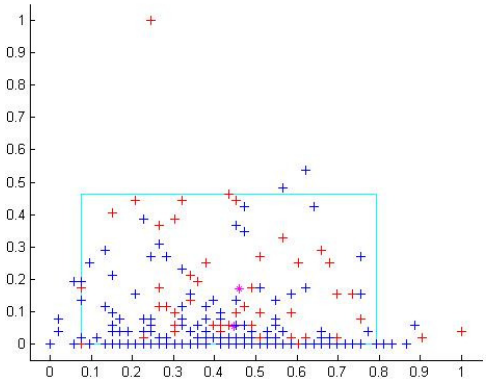


En rojo el cubo y los datos con salida 1 los límites del cubo están marcados por los datos con salida 1.

En azul el cubo y los datos con salida 0 los límites del cubo están marcados por los datos con salida 0.

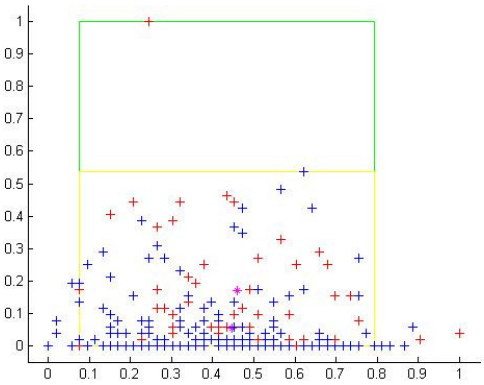
En rosa la intersección de los dos cubos, con salida en función de la distancia a los dos centroides, marcados en la imagen con * de color rosa.

Segunda Iteración



En cyan el único cubo obtenido en esta iteración, salida 1. Surge al calcular los cubos pertenecientes a la intersección de los dos cubos iniciales (cubo rosa), al hacer la división surgen 2 cubos nuevos, uno de dimensiones semejantes al cubo rosa, por lo que lo descartamos, el otro es el cubo cyan.

Tercera Iteración



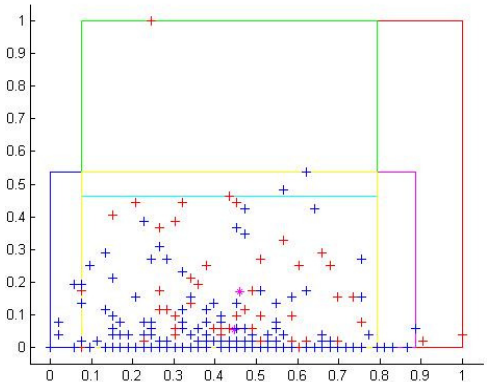
En la tercera y última iteración que permitimos a la aplicación surgen dos cubos más. Surgen a partir de los cubos iniciales, realizan una división de los mismos eliminando algunos de los casos.

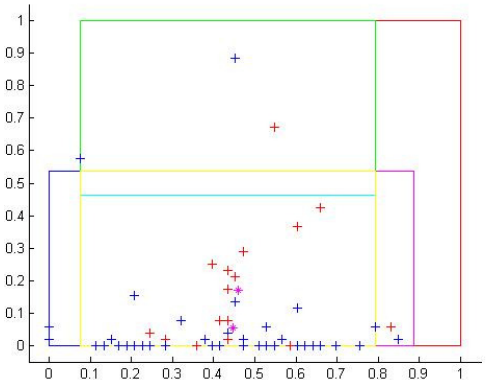
El cubo de color verde tiene salida 1.

El cubo de color amarillo tiene salida 0

Disposición Final de los Cubos

Esta es la representación de cómo quedarían los distintos cubos finalmente representando en el primer gráfico los datos de Train y en el segundo diagrama representamos los cubos junto con los datos de Test.





2.2.3 - Resultados detallados para el dataset “Haberman”

Para explicar detalladamente el proceso en un ejemplo sencillo hemos tenido en cuenta el dataset Haberman, en su primera partición. Los valores dados a los umbrales han sido Th1 = 0.5 y Th2 = 0.6. Los hipercubos que obtenemos son los representados en la tabla siguiente:

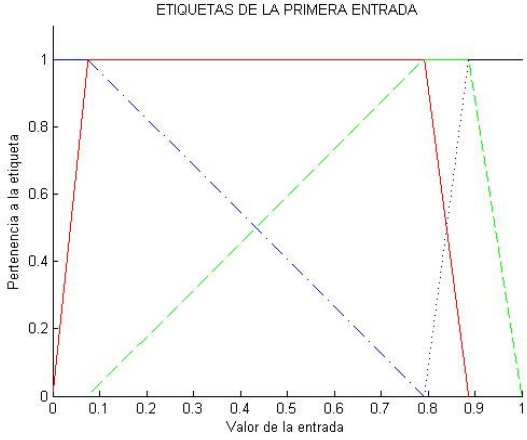
CUBOS	ENTRADA 1	ENTRADA 2	ENTRADA 3	SALIDA
C1	0.0755	0	0	**1**
	1	1	1	**1**
C2	0	0	0	**0**
	0.8868	1	0.5385	**0**
C3	0.0755	0	0	**-1**
	0.8868	1	0.5385	**-1**
C4	0.0755	0	0	**1**
	0.7925	1	1	**1**
C5	0.0755	0	0	**1**
	0.7925	1	0.4615	**1**
C6	0.0755	0	0	**0**
	0.7925	1	0.5385	**0**

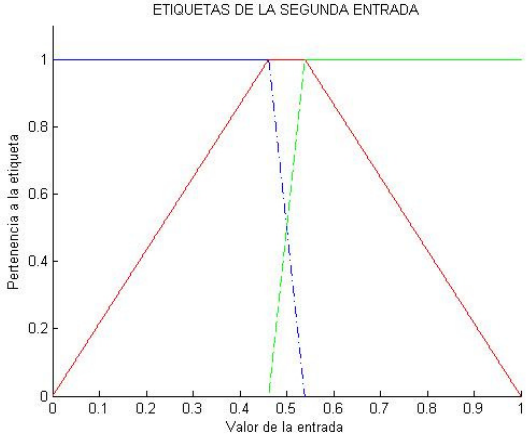
Como se observa en la tabla anterior, la segunda entrada no aporta ninguna información para el cálculo del resultado, ya que sea cual sea su valor no afectará de ningún modo a la pertenencia o no a alguno de los cubos, porque todos ellos aceptan todo el rango de la segunda entrada.

Para cada una de las entradas vamos a etiquetar los distintos valores que puede tomar para de esta forma poder crear fácilmente las reglas.

ENT.1 VALORES	ENT.1 ETIQUETAS		ENT.3 VALORES	ENT.3 ETIQUETAS
0 – 0.0755	Low		0 – 0.4615	Low
0.0755 – 0.7925	Low-Med		0.4615 – 0.5385	Med
0.7925 – 0.8868	Med-High		0.5385 – 1	High
0.8868 - 1	High			

En las gráficas siguientes observamos las funciones de pertenencia a cada una de las etiquetas de las dos entradas significativas en esta partición del dataset.





Las reglas se obtienen "traduciendo" los límites de cada cubo a etiquetas lingüisticas, es decir, estudiaremos cuales son los límites de cada cubo para colocar, en cada antecedente la etiqueta que le corresponda. Se crean tantas reglas como cubos se obtienen de los datos recibidos.

Regla	ENTRADA 1	ENTRADA 3	CLASS	Cubo
R1	Low-Med	Low	1	5
R2	Low-Med	Med	0	6
R3	Low-Med	High	1	4
R4	Med-High	Low ó Med	0 ó 1 (Dist)	3
R5	Low	Low ó Med	0	2
R6	High	Low ó Med ó High	1	1

Para la obtención de la salida de la R4 habrá que tener en cuenta la distancia del dato recibido a cada uno de los centroides, y la menor distancia determinará la salida del caso. Para ello, creamos una tabla con los centroides, en la cual, si que nos interesará el valor de la entrada 2 para calcular la distancia de cada dato a cada centroide y quedarnos con la mínima.

CENTROIDE	ENTRADA 1	ENTRADA 2	ENTRADA 3	CLASS
Centroide 1	0.4606	0.4652	0.1708	1
Centroide 2	0.4471	0.4165	0.0532	0

Con estas reglas obtenidas, aplicamos la inferencia sobre el total de los datos del dataset, tanto en el conjunto de train como en el de test y obtenemos los siguientes resultados:

CONJUNTO	ACIERTOS	TOTAL	% ACIERTOS	NUM DE REGLAS
Train	185	244	75.8197 %	6
Test	45	62	72.5806 %	6

2.2.4 – Resultados Obtenidos

En este apartado se muestran los resultados obtenidos en cada uno de los conjuntos de datos de que disponemos, en total son 21 conjuntos. Cada conjunto de datos consta de 5 particiones, por lo tanto se efectuarán 5 pruebas con cada uno de ellos, en cada prueba, el conjunto de test será uno diferente.

Los resultados mostrados para cada conjunto son los siguientes:

Train – Porcentaje de acierto en la clasificación de los datos de entrenamiento.

Test – Porcentaje de acierto en la clasificación de los datos de test.

Reglas – Número de reglas obtenidos para la clasificación de los datos del conjunto.

En la última fila, se muestra la media de los resultados de los conjuntos de train, test y el número de reglas medio que se obtiene en ese conjunto de datos.

WISCONSIN			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	92.3077	93.4307	4
2	89.5604	88.3212	2
3	90.8425	91.2409	2
4	88.6654	91.9118	3
5	89.7623	87.5	2
MEDIA	**90.22766**	**90.48092**	**2.6**

BUPA			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	64.1304	57.971	8
2	64.4928	59.4203	8
3	64.1304	60.8696	8
4	52.1739	47.8261	8
5	63.7681	62.3188	8
MEDIA	**61.73912**	**57.68116**	**8**

PIMA			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	68.8925	70.1299	8
2	69.8697	65.5844	8
3	70.3583	65.5844	8
4	71.8699	67.3203	8
5	69.5935	66.0131	8
MEDIA	**70.11678**	**66.92642**	**8**

HABERMAN			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	74.5902	74.1935	6
2	74.6939	73.7705	6
3	75.9184	73.7705	7
4	32.2449	24.5902	4
5	75.5102	73.7705	6
MEDIA	**66.59152**	**64.01904**	**5.8**

VEHICLE			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	78.9941	74.7059	4
2	79.1728	73.9645	4
3	77.9911	77.5148	4
4	77.8434	78.1065	4
5	78.582	74.5562	4
MEDIA	**78.51668**	**75.76958**	**4**

BALANCE			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	46	46.4	2
2	46	46.4	2
3	46	46.4	2
4	46	46.4	2
5	46	46.4	2
MEDIA	**46**	**46.4**	**2**

CLEVELAND			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	62.0253	45	7
2	54.4304	55	7
3	60.084	50.8475	7
4	56.3025	45.7627	7
5	55.4622	49.1525	7
MEDIA	**57.66088**	**49.15254**	**7**

ECOLI			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	72.3881	64.7059	5
2	87.7323	94.0299	7
3	57.6208	56.9231	4
4	90.7063	91.0448	5
5	95.539	88.0597	4
MEDIA	**80.7973**	**78.95268**	**5**

GLASS			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	79.5322	72.093	7
2	79.5322	76.7442	7
3	82.4561	69.7674	7
4	73.6842	79.0698	5
5	84.8837	76.1905	8
MEDIA	**80.01768**	**74.77298**	**6.8**

HAYES-ROTH			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	75.2381	59.2593	3
2	70.4762	70.3704	3
3	71.6981	69.2308	3
4	71.6981	65.3846	3
5	70.7547	65.3846	3
MEDIA	**71.97304**	**65.92594**	**3**

IRIS			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	100	100	2
2	100	100	2
3	100	100	2
4	100	100	2
5	100	100	2
MEDIA	**100**	**100**	**2**

MAGIC			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	38.1328	34.6457	9
2	38.4615	35.4331	8
3	37.7135	34.4737	8
4	38.3706	36.0526	9
5	37.9106	37.1053	8
MEDIA	**38.1178**	**35.54208**	**8.4**

NEW-THYROID			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	94.7674	93.0233	3
2	97.093	97.6744	3
3	94.186	90.6977	3
4	97.6744	100	3
5	92.4419	97.6744	3
MEDIA	**95.23254**	**95.81396**	**3**

PAGEBLOCKS			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	50.6849	40.9091	8
2	54.3379	45.4545	8
3	73.5160	72.7273	8
4	58.0866	58.7156	7
5	99.0888	94.4954	6
MEDIA	**67.14284**	**62.46038**	**7.4**

PENBASED			
PARTICIÓN	TRAIN	TEST	REGLAS
1	87.3864	87.2727	4
2	84.5455	81.3636	4
3	79.4318	82.2727	4
4	93.7500	90.9091	4
5	81.3636	83.1818	4
MEDIA	**85.29546**	**84.99998**	**4**

SEGMENT			
PARTICIÓN	TRAIN	TEST	REGLAS
1	98.1061	95.4545	4
2	98.1061	96.3203	3
3	96.4827	95.4545	3
4	96.1039	95.2381	3
5	96.3203	95.8874	2
MEDIA	**97.02382**	**95.67096**	**3**

SHUTTLE			
PARTICIÓN	TRAIN	TEST	REGLAS
1	86.3793	88.2488	8
2	98.9655	99.0805	10
3	99.0805	97.4713	10
4	87.0115	85.977	4
5	99.2529	97.931	11
MEDIA	**94.13794**	**93.74172**	**8.6**

TWONORM			
PARTICIÓN	TRAIN	TEST	REGLAS
1	97.9730	93.9189	6
2	97.9730	87.1622	6
3	98.3108	90.5405	9
4	96.2838	87.8378	6
5	97.9730	89.1892	6
MEDIA	**97.70272**	**89.72972**	**6**

WINE			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	100	97.2222	3
2	100	86.1111	4
3	100	88.8889	3
4	100	85.7143	3
5	100	91.4286	3
MEDIA	**100**	**89.87302**	**3.2**

YEAST			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	84.1618	83.1650	8
2	84.3302	83.1650	8
3	83.5720	83.5017	5
4	84.2460	82.4916	8
5	31.1448	28.7162	8
MEDIA	**73.49096**	**72.2079**	**7.4**

RING			
PARTICIÓN	**TRAIN**	**TEST**	**REGLAS**
1	93.7500	82.4324	5
2	95.7770	87.1622	3
3	87.1622	79.7297	5
4	98.9865	87.5862	5
5	98.9865	94.4828	4
MEDIA	**94.93244**	**86.27866**	**4.4**

A continuación se muestra una tabla resumen de los conjuntos de datos con el porcentaje medio de acierto de cada uno de ellos junto con el número medio de reglas generadas para cada conjunto de datos.

Para cada conjunto de datos se muestra la media de aciertos entre el conjunto de datos de entrenamiento y el conjunto de test, así como el número medio de reglas entre las distintas particiones del dataset. En la última fila de la tabla se muestra también la media total de porcentaje de aciertos con el método y el número medio de reglas entre todos los conjuntos de datos estudiados.

RESUMEN		
CONJUNTO DE DATOS	**% DE ACIERTO**	**NÚMERO DE REGLAS**
WISCONSIN	90.48092 %	2.6
BUPA	57.68116 %	8
PIMA	66.92642 %	8
HABERMAN	64.01904 %	5.8
VEHICLE	75.76958 %	4
BALANCE	46.4 %	2
CLEVELAND	49.15254 %	7
ECOLI	78.95268 %	5
GLASS	74.77298 %	6.8
HAYES-ROTH	65.92594 %	3
IRIS	100 %	2
MAGIC	35.54208 %	8.4
NEW-THYROID	95.81396 %	3
PAGEBLOCKS	62.46038 %	7.4
PENBASED	84.99998 %	4
SEGMENT	95.67096 %	3
SHUTTLE	93.74172 %	8.6
TWONORM	89.72972 %	6
WINE	89.87302 %	3.2
RING	86.27866 %	4.4
YEAST	72.2079 %	7.4
MEDIA	**75.06665 %**	**5.22**

En la tabla de la página siguiente se compara el método con otros cuatro métodos de clasificación como son el C4.5, Chi, FH-GBML y FID-3 para observar a que nivel se encuentra el método estudiado en este apartado con respecto de las alternativas existentes actualmente.

Para cada conjunto de datos se ha remarcado en negrita el método con el que se obtiene el mayor número de aciertos.

COMPARACIÓN DE MÉTODOS (% DE ACIERTO)					
CONJUNTO DE DATOS	**MÉTODO ESTUDIADO**	**C 4.5**	**CHI**	**FH-GBML**	**FID-3**
BALANCE	46,4 %	77,28 %	89,92 %	77,6 %	**90,08 %**
BUPA	57,68116 %	**66,08696 %**	57,68116 %	63,47826 %	58,55076 %
CLEVELAND	49,15254 %	51,82486 %	36,0113 %	**54,55367 %**	51,48588 %
ECOLI	**78,95268 %**	78,28358 %	72,63828 %	72,90606 %	76,49256 %
GLASS	**74,77298 %**	68,72647 %	57,95127 %	57,48616 %	53,77632 %
HABERMAN	64,01904 %	72,22105 %	72,87678 %	71,89318 %	**73,20464 %**
IRIS	**100 %**	93,33333 %	92,66667 %	95,33333 %	94,66666 %
MAGIC	35,54208 %	**79,81061 %**	74,86738 %	78,49441 %	77,44288 %
NEW-THYROID	**95,81396 %**	91,16279 %	84,65116 %	92,55814 %	90,23256 %
PAGEBLOCKS	62,46038 %	**95,06589 %**	91,42118 %	94,1568 %	92,6956 %
PENBASED	84,99998 %	89,36364 %	94,27273 %	67,18182 %	**95,1818 %**
PIMA	66,92642 %	74,08624 %	72,52695 %	73,95892 %	**76,42814 %**
RING	**86,27866 %**	82,7027 %	52,7027 %	82,83784 %	51,4865 %
WINE	89,87302 %	94,90476 %	92,66667 %	90,96825 %	**97,14286 %**
WISCONSIN	90,48092 %	95,02576 %	90,48519 %	95,60863 %	**96,34392 %**
MEDIA	**72,223 %**	**80,658 %**	**75,556 %**	**77,934 %**	**78,347 %**

2.3 - APLICACIÓN DEL MÉTODO A LA CLASIFICACIÓN DE PÍXELES EN IMÁGENES

2.3.1 – Descripción del Método

2.3.1.1 - Procesamiento Inicial

2.3.1.1.1– Procesamiento Previo

Lo primero que se ha realizado en esta aplicación es realizar un preprocesado de los datos de entrada, ya que los recibidos inicialmente eran bastante caóticos y no seguían un esquema general. Para ello se ha creado un fichero de información sobre los archivos de cada dataset, que proporcionará información sobre el número de imágenes de cada uno, el nombre de las mismas y la referencia de la imagen "ground truth". En la Fig. 1 se observa una de las imágenes de entrada y en la Fig. 2 se observa el "ground truth" de dicho dataset

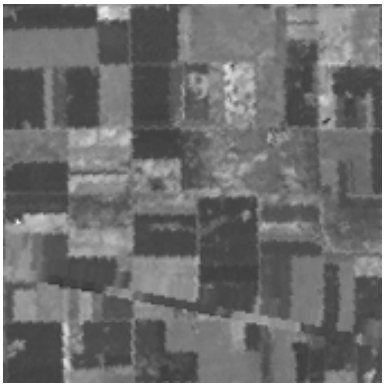


Fig. 1

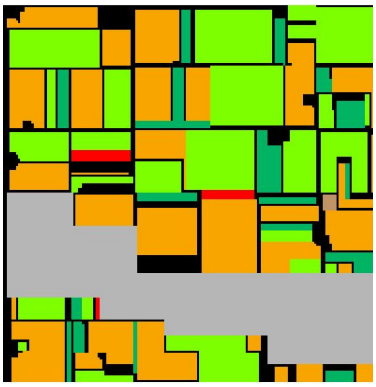


Fig. 2

2.3.1.1.2– Importar Datos a Matlab

Se ha creado la función "getInfo", cuyo objetivo es la lectura del archivo de información del dataset que se está trabajando y así facilitar el trabajo de la función principal una vez conoce el número y los nombres de las imágenes de que consta el dataset.

2.3.1.1.3– Adecuación de la Imagen Ground Truth

La imagen ground truth que se recibe es una imagen a color, con un número de tonalidades desconocido, dado que cada tonalidad representará una clase distinta, se deberá conocer el número de clases existentes en el dataset. Para obtener esta

información se han creado dos funciones para el trabajo exclusivo con la imagen ground truth.

2.3.1.1.3.1 – CONTAR NIVELES

Se ha creado una función con el nombre contarNiveles, cuyo objetivo es el de contar el número de tonalidades distinas que se tienen en una imagen ground truth, en realidad, la salida que proporciona se corresponde con el número de salidas distintas que tiene cada dataset, ya que cada nivel de gris de la imagen se corresponde con una tonalidad.

2.3.1.1.3.2 – PREPARAR GROUND TRUTH

Una vez se tienen el número de clases de un dataset se llamará a esta función, la cual devolverá una matriz del tamaño de la imagen Ground Truth que proporcionará información sobre la clase a la que pertenece cada píxel de la imagen, olvidándose de colores pasamos a trabajar con clases numeradas comenzando por 0 hasta el número de clases del dataset. En la Fig. 3 se observa la imagen recibida como "ground truth", y en la Fig. 4 se observa la matriz de clases (tratada de forma que el ojo humano pueda apreciar bien los distintos niveles).

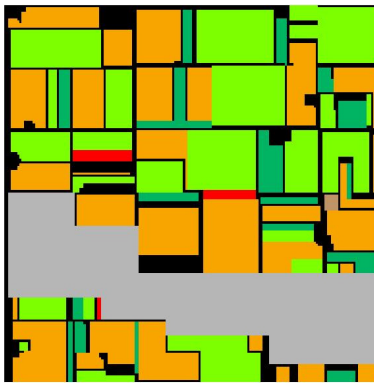


Fig. 3

Fig. 4

2.3.1.1.4 – PROCESAMIENTO DE LAS IMÁGENES DE ENTRADA

Las imágenes de entrada son un conjunto de imágenes en escala de grises, las cuales no abarcan la totalidad de intensidades que el rango de la escala de grises ofrece, hay imágenes sobreexpuestas e imágenes subexpuestas, por lo que se va a realizar un tratamiento previo de las entradas y se van a normalizarlas para usar la totalidad del

rango del que se dispone. Antes de normalizarlas, y con la finalidad de obtener un buen normalizado, se va a eliminar el ruido de la imagen.

2.3.1.1.4.1 – ELIMINACIÓN DE RUIDO EN LA IMAGEN

Se va a eliminar el ruido existente en la imagen mediante la utilización del filtro de la media. Se pasará una máscara de 3x3 sobre la imagen y de esta forma se eliminará el ruido mediante un suavizado general de la imagen. En la Fig. 5 se observa la imagen sin tratar y en la Fig. 6 la imagen sin ruido.

Fig. 5

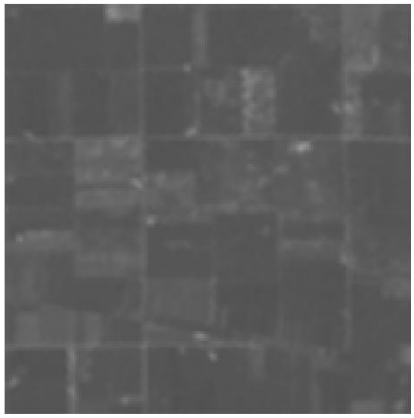


Fig. 6

2.3.1.1.4.2 – NORMALIZACIÓN DE LA IMAGEN

Mediante la normalización de la imagen lo que se pretende es la utilización del total del rango de intensidades de la escala de gris, de esta forma se distinguirán mejor los píxeles de un área con respecto a los píxeles de otro área. También conseguiremos que todas las imágenes se utilicen el mismo rango, ya que de otra forma puede producirse que una imagen utilice, por ejemplo, el rango 0-150 y otra utilice el 120-255 de forma que no se asociarán unas tonalidades de una imagen con las mismas tonalidades de la otra.

Con éste paso se consiguen los resultados mostrados en las siguientes imágenes, en la Fig. 7 se muestra la imagen sin normalizar, y en la Fig. 8 se muestra la imagen normalizada.

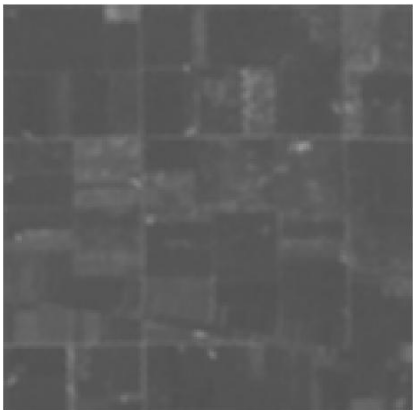


Fig. 7

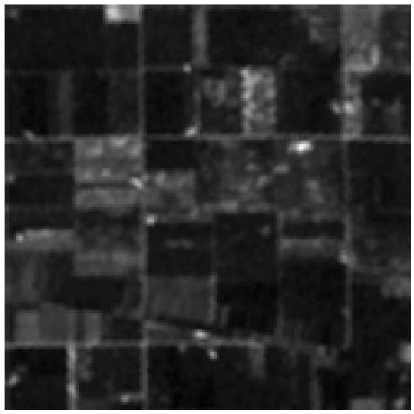


Fig. 8

2.3.1.1.5 – FIN PROCESAMIENTO INICIAL

El último paso del procesamiento inicial consiste en guardar estas imágenes en un directorio temporal que se va a crear desde la propia herramienta llamado "tmp_Normalizadas" en el cual se guardarán las imágenes normalizadas pero con el mismo nombre que en sus carpetas originales. A partir de este momento se trabajará con esta carpeta temporal en vez de trabajar con las imágenes de entrada.

Al finalizar la ejecución de la aplicación, y con la finalidad de no dejar rastro de la ejecución de la misma en el ordenador se borrará la carpeta temporal así como todas las imágenes que contenga.

2.3.1.2 – OBTENCIÓN DE CONJUNTOS Y CÁLCULO DE SUS SALIDAS

El método escrito en el artículo *'An interpretable fuzzy rule-based classification methodology for medical diagnosis'* describe el proceso de obtención de reglas basándose en hipercubos ya que el número de variables existentes en ese dataset es variable. En el caso del tratamiento de imágenes tan solo se trabaja en una dimensión, la que representan los niveles de gris de los píxeles de la imagen.

Se va a trabajar mediante intervalos, que a efectos prácticos darán los mismos resultados que se obtienen con los cubos, pero con una sola dimensión.

Se comenzará trabajando con intervalos no difusos para después fuzzificarlos una vez se tengan todos los intervalos definitivos creados.

2.3.1.2.1 – OBTENCIÓN DE LOS INTERVALOS INICIALES

Se va a crear una función que calcule los intervalos iniciales obtenidos por los valores de gris de cada píxel en el total de las imágenes.

Esta función proporcionará también una matriz que representará mediante unos y ceros los valores que se toman como entrada y los valores que se toman como test. Estos valores serán tomados de forma aleatoria mediante una función que hará seleccionar el 50% de los valores como train y el otro 50% como test.

Al ejecutar la función con el dataset Tipjul1 se obtienen unos conjuntos iniciales similares a los mostrados en la Tabla 1, en función de los datos de train y los datos de test seleccionados. Estos conjuntos serán fuzzificados más adelante, ya que primero se tienen que estudiar las intersecciones entre ellos, que son abundantes.

	NIVEL 0	NIVEL 1	NIVEL 2	NIVEL 3	NIVEL 4	NIVEL 5	NIVEL 6
MIN	4	29	7	18	0	0	2
MAX	255	159	244	143	223	214	253

Tabla 1

A continuación se muestra, en la Fig. 9 la gráfica en la cual representamos como quedarían los distintos intervalos obtenidos, en ella se ve claramente como hay gran cantidad de intersecciones que se deberán estudiar.

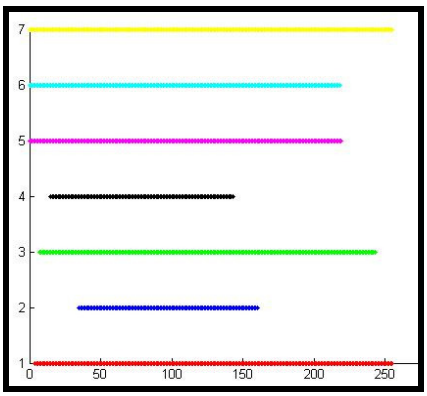


Fig. 9

2.3.1.2.2 – CÁLCULO DE INTERSECCIONES ENTRE INTERVALOS

Una vez se tienen creados los distintos intervalos se deberán estudiar las intersecciones entre ellos para determinar los distintos intervalos que se crean, teniendo en cuenta las distintas salidas asociadas que pueden tener cada intervalo.

En el ejemplo de los intervalos anteriores, los nuevos intervalos que se crean son los mostrados en la Tabla 2 junto con las posibles salidas.

MIN	MAX	SALIDA
0	2	4 o 5
3	4	4 o 5 o 6
5	7	0 o 4 o 5 o 6
8	18	0 o 2 o 4 o 5 o 6
19	29	0 o 2 o 3 o 4 o 5 o 6
30	143	0 o 1 o 2 o 3 o 4 o 5 o 6
144	159	0 o 1 o 2 o 4 o 5 o 6
160	214	0 o 2 o 4 o 5 o 6
215	223	0 o 2 o 4 o 6
224	244	0 o 2 o 6
245	253	0 o 6
254	255	0

Tabla 2

Visto esto parece que el único intervalo que tiene salida asociada es el intervalo [254 - 255], que parece pertenecer a la clase 0. El resto del rango de gris pertenece como mínimo a dos intervalos, por lo que habrá que estudiar las intersecciones y obtener la salida asociada a cada intervalo o las posibles subdivisiones de los intervalos para ver si merece mas la pena la división del intervalo en subintervalos con distintas salidas.

Se ha creado para este fin la función “todos_intervalos”, recibirá los intervalos iniciales y dará como resultado el total de intervalos que surgen de las intersecciones entre ellos.

2.3.1.2.3 – CÁLCULO DEL UMBRAL R_{IJ} Y DIVISIÓN DE INTERVALOS SEGÚN ESTE VALOR

Una vez se hayan calculado los distintos intervalos surgidos por las intersecciones entre intervalos se deberá calcular el conocido como umbral R_{ij}, que determinará si se debe subdividir dicho intervalo en mas intervalos o si es suficiente con dejar el intervalo como está.

Este umbral se calculará de la misma forma que se calculaba en el caso del artículo original. La intención del mismo es mirar el porcentaje de datos que pertenecen al subintervalo con respecto al total de datos de los intervalos anteriores que han dado lugar a este subintervalo.

Para el cálculo del umbral R_{ij} aplicaremos la siguiente fórmula:

$$R_{ij} = \frac{D(A_i^{\,l} \cap A_j^{\,l})}{D(A_i^{\,1} \cup A_j^{\,1})}$$

Donde D(x) denota el número de datos pertenecientes a x.

La idea de este punto es seguir el siguiente pseudocódigo para crear todos los intervalos que quedaran finalmente. En nuestra herramienta hemos aplicado el pseudocódigo siguiente, además de algunas mejoras de programación para intentar hacerlo más rápido.

Tras la aplicación del pseudocódigo se tiene lista la información necesaria para el siguiente punto de la aplicación, que consiste en asociar a cada intervalo una salida.

La salida de esta función será una matriz de 2 filas y X columnas, en cada columna se almacenará la información de un intervalo, la primera fila contendrá el mínimo y la segunda fila contendrá el máximo de los límites de intervalo. En posteriores funciones se asociará a cada intervalo una salida, que se colocará en una tercera fila de la matriz de forma que se tendrá una matriz del estilo de la mostrada en la Tabla 3.

```
   Se estudian los intervalos recibidos
Para cada intervalo
        Calculo Rij
        Si Rij≥Th1          Se subdivide el intervalo y se colocan los subintervalos en 'a estudiar'
        [] Rij<Th1          Se envía el intervalo a 'intervalos finales'
        Fin si
Fin Para
   Se estudian los intervalos en 'a estudiar' hasta que 'a estudiar' este vacío
Mientras 'a estudiar' no esté vacío
        Calculo Rij
        Si Rij<Th1          Se subdivide el intervalo y se colocan los subintervalos en 'a estudiar'
        [] Rij≥Th1          Se envía el intervalo a 'intervalos finales'
        Fin si
Fin Mientras
   Se prepara la salida
Se reordena intervalos finales
Se devuelve intervalos finales
```

Pseudocódigo

Intervalo 1	Intervalo 2	Intervalo 3		Intervalo N
Min I_1	Min I_2	Min I_3		Min I_N
Max I_1	Max I_2	Max I_3	…	Max I_N
Salida I_1	Salida I_2	Salida I_3		Salida I_N

Tabla 3

2.3.1.2.4 – CÁLCULO DE LAS SALIDAS DE CADA INTERVALO

Para calcular la salida asociada a cada intervalo se van a seguir los pasos establecidos en el artículo *'An interpretable fuzzy rule-based classification methodology*

for medical diagnosis', por lo tanto se tendrá que calcular el umbral S_{ij}, el cual indicará que método se debe seguir para el cálculo de la salida.

2.3.1.2.4.1 – CÁLCULO DEL VALOR S_{IJ}

Existen dos métodos para el cálculo de la salida asociada a cada intervalo, el criterio de densidad, que se basa en asociar a cada intervalo el mayor número de datos de una clase en el intervalo, y el criterio de distancia, que se basa en la distancia de un dato al centroide de los datos de esa clase, se asociará la menor de las distancias.

Como ya se ha mencionado, para saber que método se debe seguir para el cálculo de la salida se deberá calcular el umbral S_{ij} con anterioridad, para ello se debería aplicar la siguiente fórmula.

$$S_{ij}^{I} = \frac{|D_i(A_i \cap A_j) - D_j(A_i \cap A_j)|}{D(A_i \cap A_j)}$$

Donde D (x) denota el número de datos pertenecientes al intervalo x y Di(x) denota el número de datos de la clase i pertenecientes al intervalo x.

Pero surge el problema de que esta fórmula esta preparada para intersecciones entre tan solo 2 clases, tal y como se supone en el artículo que nos está sirviendo como base. Se deberá modificar esta fórmula de forma que tenga en cuenta que se está trabajando con varias clases, no solo con dos.

La intención de la fórmula para el caso de tener intersecciones entre tan solo dos clases es mirar si el número de datos existentes en la intersección es, en su mayoría, de una de las dos clases, para, en ese caso, aplicar el criterio de densidad (cuando el valor S_{ij} es mayor o igual que el umbral Th2 que sea indicado por el usuario).

Se ha pensado solucionar este tema calculando cual es la clase que mayor representación tiene dentro del intervalo, el valor S_{ij} que se va a suponer será el porcentaje de datos de la clase con mayor representación dentro del intervalo. De esta forma el valor S_{ij} va a tener la misma intención que en el artículo y será calculado de una forma similar, por lo que guardarán cierta relación entre ambos valores.

2.3.1.2.4.2 – CRITERIO DE DENSIDAD

El criterio de densidad asocia a un intervalo la salida mas frecuente dentro de los datos que pertenezcan a dicho intervalo. Para ello calcula el peso de cada clase dentro del intervalo y mirará cual es la clase con mayor peso para considerar esa clase como salida del intervalo.

Para calcular el peso de una clase se aplica la siguiente fórmula que tiene en cuenta los datos de la clase y los datos totales

$$W_i = \frac{D_i(A_i)}{D(A_i)}$$

Donde D(x) es el número de datos pertenecientes al cubo x y $D_i(x)$ son el numero de datos de la clase i pertenecientes al cubo x.

Según el artículo del que se parte, las reglas que se obtienen en este paso son del tipo:

"IF x is in A_{ij} THEN y is in Y_i WHEN $W_i>W_j$, OR y is in Y_j WHEN $W_i<W_j$"

No obstante al estar trabajando con varios tipos de clase, no solo 2, se tendrán reglas del tipo

"IF x is in A_i THEN y is in Y_i WHEN ($W_i>W_j$ and $W_i>W_k$ and...), y is in Y_j WHEN ($W_j>W_i$ and $W_j>W_k$ and...), y is in Y_k WHEN ($W_k>W_i$ and $W_k>Wj$ and...),..."

Parece mas complicado de lo que en realidad va a terminar siendo, ya que a la hora de programarse existirá un único intervalo con una única salida, ya que los pesos serán calculados con anterioridad a realizar la inferencia. Por lo que finalmente se obtendrá una regla del tipo:

"IF x is in A_i THEN y is in Y_i"

Y la salida Y_i será la salida asociada a la case con mayor peso en el intervalo A_i por lo que tendrá el mismo efecto a la hora de realizar la inferencia que las reglas anteriores.

2.3.1.2.4.3 – CRITERIO DE DISTANCIA

El criterio de distancia intenta observar la distribución de los datos de cada clase dentro del intervalo, para calcular la salida en función de la proximidad a cada conjunto de datos del dato de entrada. Para ello se calcula el punto central de los datos de cada clase, y de esta forma se obtiene el centroide de la clase.

Para calcular el centroide de una clase se debe seguir la siguiente fórmula, que tiene en cuenta el valor asociado a cada punto de la clase.

$$C(k) = \frac{x_1(k) + x_2(k) + ... + x_N(k)}{N}$$

La fórmula anterior está pensada para cubos, en este caso, al tratar con tan solo una dimensión no será necesario mas que calcular la media de todos los datos de cada clase, es decir, lo que se hace para cada dimensión del cubo. Una fórmula 'ad hoc' para este caso será la siguiente:

$$C = \frac{x_1 + x_2 + ... + x_N}{N}$$

Según el artículo, una vez se tienen calculados los centroides, se deberá calcular la distancia de un dato a cada uno de los centroides siguiendo la fórmula de la distancia euclídea, representada por la siguiente fórmula:

$$d_i = \sqrt{\left(x_1 - x_1^{Ci}\right)^2 + \left(x_2 - x_2^{Ci}\right)^2 + ... + \left(x_m - x_m^{Ci}\right)^2}$$

En este caso tenemos una sola dimensión, por lo que calcular la distancia euclídea no es necesaria, calcularemos una distancia simple entre datos que será la representada por la siguiente fórmula:

$$d_i = \left| x_i - \text{Centroide} \right|$$

Al igual que en el artículo original, las reglas obtenidas con este método serán del tipo:

"IF x is in A THEN y is in Y_i WHEN (d_j>d_i and dk>di and...)OR y is in Y_j WHEN (d_i>d_j and d_k>d_j and...) OR y is in Y_k WHEN (d_i>d_k and d_j>d_k and...) OR..."

2.3.1.3 – OBTENCIÓN DE RESULTADOS PROVISIONALES

Hasta este punto, este método difiere mínimamente del método original, pero es en éste momento en el que encontramos el mayor de los problemas para aplicar la solución en estos nuevos conjuntos.

El método original está pensado para el trabajo en el campo de la medicina, por lo que las posibles salidas existentes en los conjuntos serán 0 ó 1, en función de si el paciente está sano o si está enfermo. En el tratamiento en imágenes, las imágenes de satélite captan variedad de terrenos distintos, por lo tanto, el número de salidas existentes en cada conjunto se multiplica, lo cual hace que los resultados que nos da la aplicación del método original no son buenos, son mejores que un resultado aleatorio, por lo que vemos que nuestro sistema "aprende", pero no son buenos resultados.

Como muestra, en las tablas siguientes observamos como los resultados son bastante malos, hemos ido eliminando clases del total de clases existentes en el conjunto, para comprobar que, efectivamente, si reducimos el número de clases existentes, los resultados mejoran.

MODDIMENSION		
PRUEBA	**TRAIN**	**TEST**
Con todas las clases activas:		
Un total de 14 niveles	12.1063 %	12.044 %
Con una parte de las clases activas:		
Azul, Naranja, Rosa, Morado	16.5210 %	16.3859 %
Amarillo, Verde, Añil, Negro	19.3719 %	19.2229 %
Reconociendo si pertenece o no a una clase:		
Amarillo	26.1268 %	26.1281 %
Añil	65.0757 %	65.1460 %
Azul Intenso	32.0287 %	31.9688 %
Rosa	63.3182 %	63.3243 %
Negro	41.5813 %	41.6446 %
Naranja	65.3915 %	65.3703 %

TIPJUL		
PRUEBA	**TRAIN**	**TEST**
Con todas las clases activas:		
En total 6 clases diferenciables	23.932 %	23.584 %
Con una parte de las clases activas:		
Verde claro y Verde oscuro	66,9651 %	66,9652 %
Naranja y BackGround	49.1101 %	49.3377 %
Rojo y Borde	69.6024 %	69.6537 %
Reconociendo si pertenece o no a una clase:		
Naranja	70.7739 %	70.5785 %
Verde oscuro	93.8927 %	93.8098 %
Verde claro	66.7716 %	66.453 %
Rojo	99.1256 %	99.1157 %
Background	78.574 %	78.8144 %
Borde	75.2411 %	75.4267 %

Una vez observado este aspecto del método de clasificación se decidió aplicar algoritmos de aprendizaje por parejas, con los cuales nos basaremos en las clasificaciones obtenidas al reconocer tan solo una de las clases de nuestro conjunto, daremos un peso,

al valor obtenido en función de la seguridad que tengamos de que se encuentra bien clasificado.

2.3.1.4 – APRENDIZAJE POR PAREJAS

El aprendizaje por parejas que hemos realizado consiste en clasificar los píxeles de una imagen realizando varios estudios, uno por cada clase existente en la imagen, es decir, suponemos una imagen con 10 clases, pues clasificaremos toda la imagen en función de si pertenece a la clase 1 o no, a continuación clasificaremos todos los píxeles en función de si pertenecen a la clase 2 o no, luego clase 3, clase 4, etc.

Tras ese paso, tendremos cada píxel clasificado en función de si pertenece a cada una de las clases. Pero surge un problema, y es que, un mismo píxel puede ser clasificado como perteneciente a varias clases, por lo que deberemos saber con cual de las propuestas nos quedamos.

Para ello vamos a introducir el concepto de peso en cada una de las clasificaciones, con este valor, entre 0 y 1, vamos a indicar la seguridad que tenemos de la clasificación realizada, es decir, cuanto mas próximo a 1 se encuentre el peso dado a la clasificación más seguros estaremos de que la clasificación es correcta. En último lugar vamos a quedarnos con la clasificación cuyo peso sea mayor. Por ejemplo, si un píxel pertenece a la clase A con peso 0.5, a la clase B con peso 0.3 y a la clase E con peso 0.8 lo clasificaremos como perteneciente a la clase E.

2.3.1.4.1 – CÁLCULO DEL PESO DE LA CLASIFICACIÓN

Para el cálculo del peso de la clasificación de cada píxel vamos a diferenciar los píxeles clasificados mediante el criterio de densidad de los píxeles clasificados mediante el criterio de distancia.

2.3.1.4.1.1 – PESO DE LA CLASIFICACIÓN MEDIANTE EL CRITERIO DE DENSIDAD

Si se aplica el criterio de densidad es porque el porcentaje de datos pertenecientes al intervalo de la clase en la que se clasifica el dato es superior al umbral introducido. Es decir, si se introduce un umbral igual a 0.8, aquellos intervalos cuyo porcentaje de datos de una de las clases sea superior al 80% tendrán asociada como salida la pertenencia a esa clase.

Como peso de la clasificación recibirá la densidad de esa clase dentro del intervalo al que pertenece, que será, como mínimo, igual al umbral introducido, si la densidad es, por ejemplo y siguiendo con el ejemplo anterior, de un 91% de la clase C, los datos serán

clasificados como pertenecientes a la clase C con un peso de 0.91 para la comparación posterior con los pesos que le sean asociados en el estudio del resto de clases.

2.3.1.4.1.2 – PESO DE LA CLASIFICACIÓN MEDIANTE EL CRITERIO DE DISTANCIA

Si se aplica el criterio de distancia es porque los datos pertenecientes al intervalo al que pertenece el dato están repartidos de forma que ninguno de ellos sea predominante, es decir, el dato mas representado en el intervalo tiene un porcentaje de representación menor que el umbral introducido. Si seguimos con el ejemplo del umbral de 0.8, el máximo de densidad permitido para una de las clases será de 79%, si llega al 80% se aplicará el criterio de densidad.

En este caso, el peso asociado a la clasificación realizada no será la densidad de la clase, como en el ejemplo anterior, se considera que al aplicar este criterio la incertidumbre es mayor que al aplicar el criterio de densidad, por lo tanto se quiere penalizar de alguna forma este criterio, por lo tanto vamos a tener en cuenta, además de la densidad de la clase en la que se ha clasificado al píxel, la proximidad al centroide de dicha clase. Es decir, aplicaremos la siguiente fórmula:

Peso = Densidad de la Clase * Proximidad
Proximidad = 1 – Distancia al Centroide / Amplitud del Intervalo

Vamos a considerar que cuanto más cerca se encuentre el valor del centroide de la clase más probable es que la clasificación esté bien realizada, pero normalizaremos ese valor con la amplitud del intervalo, de modo que la proximidad valdrá como mínimo 0, en el caso en que el valor es un extremo del intervalo y el centroide se encuentra en el otro extremo, y como máximo 1, en el caso en que el valor sea el mismo que el centroide. Estos valores luego los penalizaremos multiplicándolos por la densidad de la clase, que, en el ejemplo puesto hasta ahora, será, como máximo 0.79.

El criterio de distancia dará siempre pesos menores que el criterio de densidad para un mismo valor del umbral, en el ejemplo del umbral 0.8, el peso del criterio de densidad se encontrará en el intervalo entre 0.8 y 1 y el peso del criterio de distancia se encontrará entre 0 y 0.79.

2.3.1.4.2 – CÁLCULO DE LA SALIDA EN FUNCIÓN DE LOS PESOS

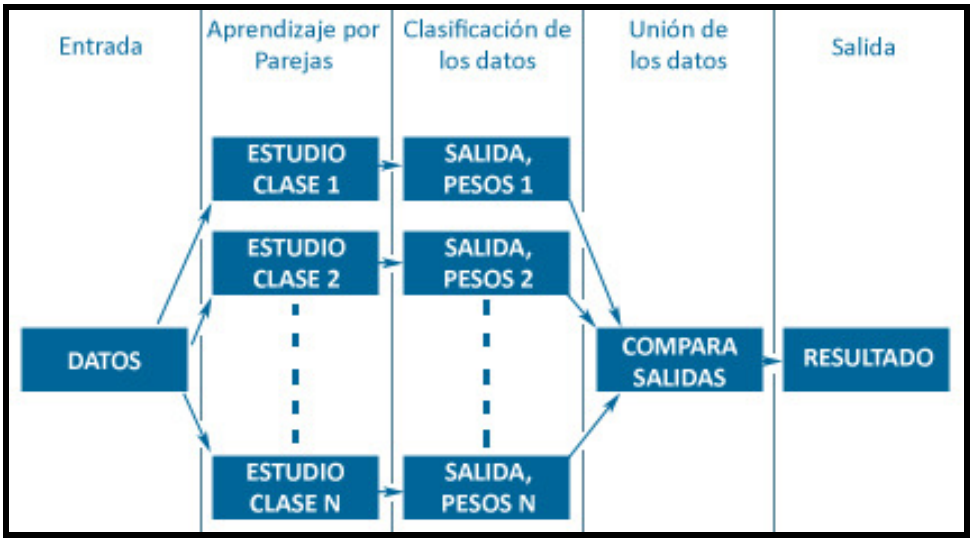
Este paso es el más sencillo, tras obtener las salidas y sus pesos asociados para cada una de las iteraciones del aprendizaje por parejas vamos a quedarnos con la que salida que mayor peso tiene, es decir, aquella que estamos más seguros de que sea la correcta. Para ello, simplemente vamos a comparar las posibles clasificaciones que hemos obtenido de un dato, pueden suceder tres situaciones:

A – Si solo se tiene una clasificación del dato, en cuyo caso daremos por clasificado el dato en esa salida, sin dar importancia al peso asociado, ya que el resto de clases no lo reconocen como pertenecientes a ellas.

B – Si se tienen varias clasificaciones del dato se deberán comprobar los pesos de las posibles clasificaciones y nos quedaremos con la clasificación cuyo peso sea la mayor

C – Si no se tiene ninguna clasificación del dato, dejaremos el dato sin asociarle una clase y lo consideraremos un dato "no clasificable".

Finalmente, quedará un esquema del método del modo siguiente, en el apartado de "Estudio clase X" es en el que se encuentra todo el método de clasificación de píxeles, pero para la explicación de la clasificación por parejas no se considera necesario explicar ese apartado en el esquema siguiente:



2.3.2 - Secuencia de Obtención de los Intervalos

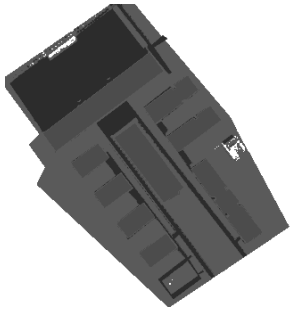
La secuencia de obtención de los intervalos dependerá de los valores que se den a los parámetros de entrada. En concreto, el valor que afectará a la creación de los intervalos es el primer umbral introducido, cuanto más restrictivo sea ese umbral, mayor número de intervalos se crearán.

Para la explicación de la creación de los intervalos vamos a utilizar unas imágenes sencillas creadas para probar el método, las imágenes utilizadas son imágenes de satélite del campus de la Universidad Pública de Navarra.

Las imágenes utilizadas son las siguientes:




La imagen ground truth con la que representamos las distintas clases reconocidas en el conjunto de datos es la siguiente:



Están representadas con distintos niveles de gris las distintas clases de interés de la imagen.

Intervalos Iniciales

Inicialmente se obtienen diversos intervalos, cada uno de ellos representa el intervalo que abarcan las intensidades de gris de los píxeles de cada clase. Los intervalos que obtenemos tienen gran número de intersecciones en común entre ellos.

Los intervalos que obtenemos en el conjunto de imágenes de la Upna son los siguientes:

	Clase 1	Clase 2	Clase 3	Clase 4
Min	0	0	0	0
Max	233	255	240	255

Segunda Iteración

Una vez calculados los intervalos iniciales observamos que en todos ellos se incluye como perteneciente a su clase el intervalo 0-237, por lo que van a surgir intersecciones, que estudiamos en esta iteración, por lo que obtendremos nuevos intervalos.

Los que obtenemos son aquellas zonas que pertenezcan a diferente número de clases, en este caso obtenemos:

	Clases 1, 2, 3 ó 4	Clases 2, 3 ó 4	Clases 2 ó 4
Min	0	234	241
Max	233	240	255

Tercera Iteración

En esta iteración vamos a tener en cuenta el umbral introducido como entrada para ver si debemos dividir el intervalo en subintervalos o si los intervalos se quedan finalmente como están.

Obtenemos los siguientes intervalos, que son los intervalos finales que vamos a utilizar para realizar nuestra clasificación:

	1, 2, 3 ó 4	1, 2, 3 ó 4	2, 3 ó 4	2, 3 ó 4	2, 3 ó 4	2 ó 4
Min	0	233	234	237	240	241
Max	232	233	236	239	240	255

Como podemos observar, el primer intervalo de la segunda iteración ha sido dividido en 2 subintervalos, y el segundo intervalo de la primera iteración ha pasado a dividirse en 3 subintervalos, mientras que el tercer intervalo se ha quedado como estaba en la iteración anterior.

En la siguiente tabla se muestra como se ha dividido cada intervalo de forma más gráfica:

Intervalos Iteración 2	Intervalos Finales
[0 – 233]	[0 – 232] [233 – 233]
[234 – 240]	[234 – 236] [237 – 239] [240 – 240]
[241 – 255]	[241 – 255]

2.3.3 – Resultados

En este apartado vamos a mostrar los resultados que obtenemos para cada conjunto de datos. La muestra original de datos se ha tomado de los conjuntos de datos utilizados en del artículo *"Type-2 Fuzzy Classifiers - Towards Interpretability"*, son un total

de 5 conjuntos de datos, formados por imágenes de satélite y una imagen que indicará, para cada conjunto de imágenes de satélite, el tipo al que pertenece cada uno de los píxeles. Además, se ha creado otro conjunto de datos más, con imágenes de satélite de la Upna modificadas y una imagen de verdad.

Para cada conjunto de datos se muestra lo siguiente:

- Imágenes de satélite
- Imagen de verdad
- Número de clases pertenecientes al conjunto de datos
- Porcentaje de acierto en el conjunto de train
- Porcentaje de acierto en el conjunto de test

Resultados Dataset MODDIMENSION

Imágenes de Satélite

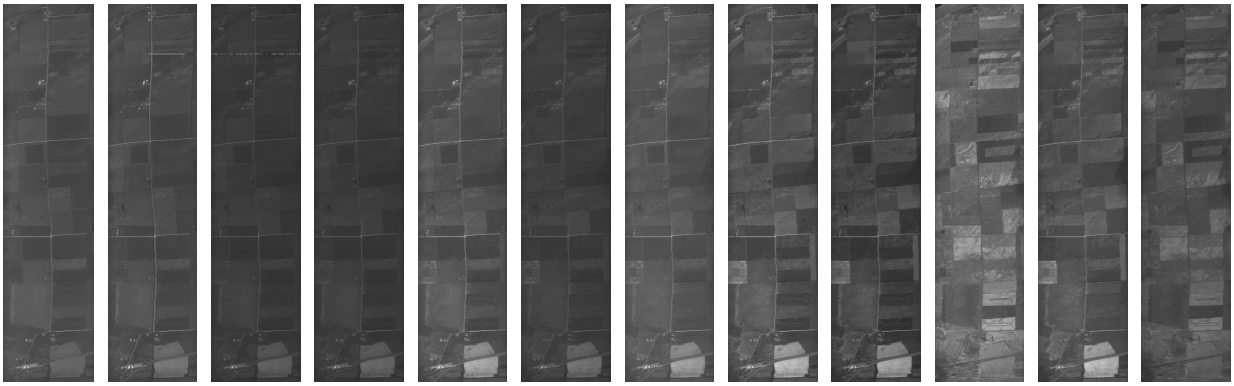


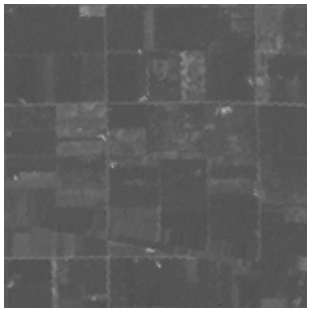
Imagen de Verdad

El dataset MODDIMENSION tiene un total de 15 clases distintas

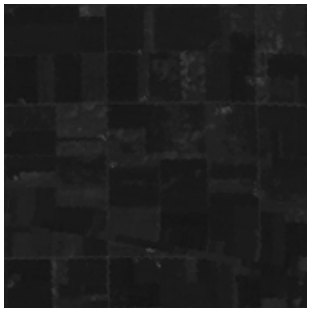
MODDIMENSION		
	TRAIN	**TEST**
Porcentaje de Acierto	25.8226 %	25.7441 %

Resultados Dataset TIPJUL

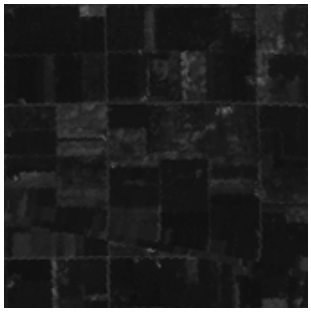
Imágenes de Satélite



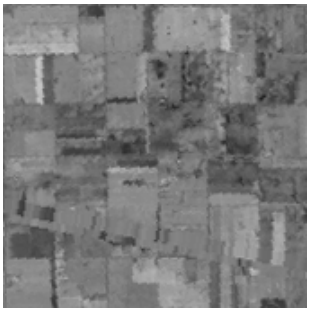
1



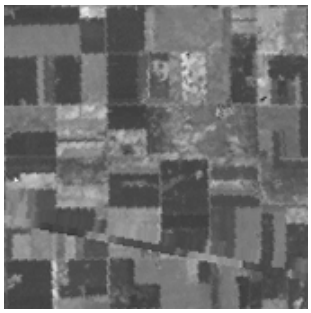
2



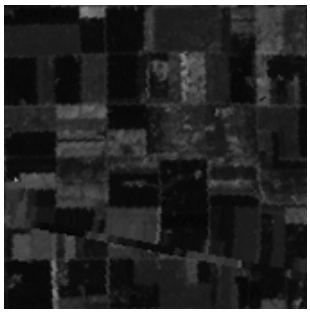
3



4



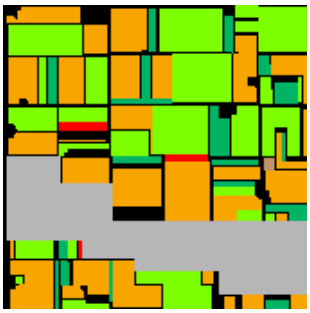
5



6

7

Imagen de Verdad



El dataset TIPJUL tiene un total de 6 clases distintas

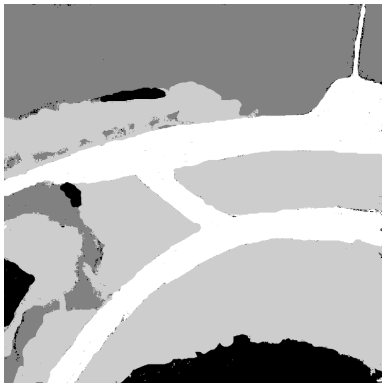
TIPJUL		
	TRAIN	**TEST**
Porcentaje de Acierto	64.8056 %	64.4502 %

Resultados Dataset Barigui1

Imágenes de Satélite

1 2 3

Imagen de Verdad

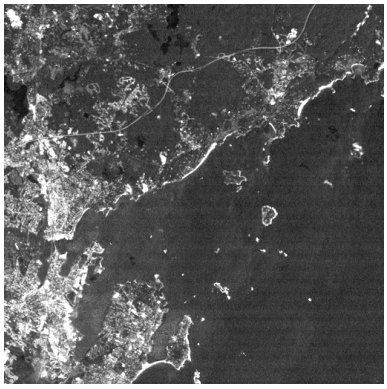


El dataset BARIGUI1 tiene un total de 4 clases distintas

BARIGUI1		
	TRAIN	**TEST**
Porcentaje de Acierto	69.9208 %	70.3196 %

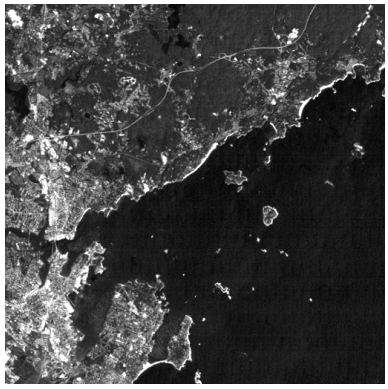
Resultados Dataset BEVERLY

Imágenes de Satélite

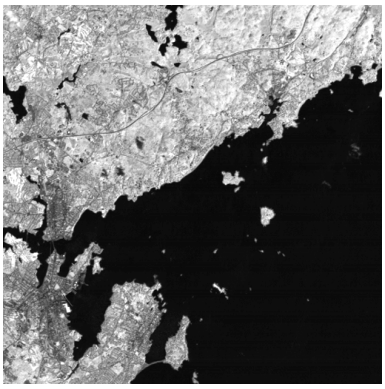


1

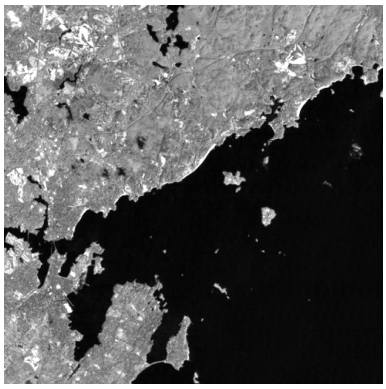
2



3

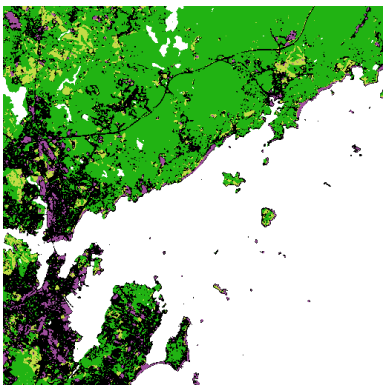


4



5

Imagen de Verdad



El dataset BEVERLY tiene un total de 5 clases distintas

BEVERLY		
	TRAIN	**TEST**
Porcentaje de Acierto	51.8355 %	51.6853 %

Resultados Dataset THYFILES

Imágenes de Satélite


1


2

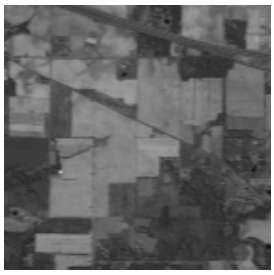

3


4


5


6


7


8


9

Imagen de Verdad

El dataset THYFILES tiene un total de 5 clases distintas

THYFILES		
	TRAIN	**TEST**
Porcentaje de Acierto	54.0955 %	54.1043 %

Resultados Dataset UPNA

Imágenes de Satélite

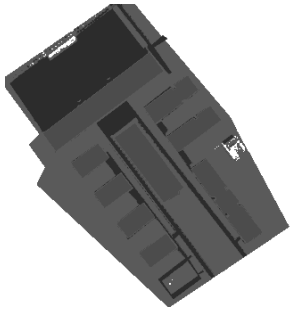

1


2


3


4

Imagen de Verdad



El dataset UPNA tiene un total de 5 clases distintas

UPNA		
	TRAIN	**TEST**
Porcentaje de Acierto	63.7820 %	63.7650 %

A continuación mostramos el resumen de los resultados obtenidos aplicando el método a la clasificación de píxeles en imágenes. Se muestra el porcentaje medio de aciertos entre el conjunto de entrenamiento y el conjunto de test para cada conjunto de datos.

RESUMEN		
CONJUNTO DE DATOS	**% DE ACIERTO**	**NÚMERO DE CLASES**
MODDIMENSION	25.7441 %	15
TIPJUL	64.4502 %	6
BARIGUI1	70.3196 %	4
BEVERLY	51.6853 %	5
THYFILES	54.1043 %	5
UPNA	63.7650 %	5
MEDIA	**55.0114 %**	**6.67**

3 - CONCLUSIONES

Tras el estudio del artículo y su aplicación con pequeñas variaciones a imágenes se obtienen las siguientes conclusiones:

3.1 – Conclusiones sobre Resultados

Teniendo en cuenta los resultados obtenidos por el método original, podemos deducir que el método es bueno para la clasificación de datos relacionados con diagnósticos médicos (paciente enfermo o paciente sano) ya que el porcentaje de acierto en el conjunto de test es bastante bueno y muy próximo a otros métodos reconocidos como buenos.

Tras observar los resultados obtenidos por la aplicación del método a imágenes observamos que los resultados son peores que los obtenidos por el método original. Esto se debe a que el número de clases existentes en los conjuntos de datos de las imágenes es superior a dos, por lo que el método original no da buenos resultados, ya que está preparado para la clasificación de datos con tan solo dos posibles salidas.

Podemos pues, concluir, que el método es bueno para conjuntos de datos con tan solo dos posibles salidas, ya que da un buen porcentaje de acierto, mientras que, para aquellos conjuntos de datos con mayor número de salidas tendremos que aplicar aprendizaje por parejas (fuera del método original) para obtener resultados buenos.

3.2 – Conclusiones sobre Configuración

En este apartado, lo que se tiene en cuenta es que el método original recibe una serie de parámetros de entrada configurables por el usuario, como son los valores de los umbrales Th1 y Th2, así como el valor de γ utilizado para fuzzificar los hipercubos obtenidos.

El problema de este tema es que el método es bastante inestable, es decir, que con pequeñas variaciones de los umbrales se pueden obtener cubos completamente distintos, y, por lo tanto, resultados muy distintos, por lo que la automatización de este método para cualquier conjunto de datos es muy complejo. Tras la realización de diversas pruebas con distintos umbrales, se ha decidido realizar todas las pruebas del método con los umbrales Th1=0.5 y Th2=0.6.

En el caso de la variación del método para su aplicación en imágenes tenemos el mismo problema a la hora de configurar los valores de los umbrales, pero en este caso desaparece el problema de configurar γ ya que no hay hipercubos, si no que se realiza el estudio mediante intervalos, por lo que no hay necesidad de fuzzificarlos ya que ocupan la totalidad del rango 0 – 255.

3.3 Conclusiones sobre Número de Reglas

Junto con los porcentajes de acierto obtenidos por el método, éste es uno de los puntos fuertes del método, debido al bajo número de reglas que genera para clasificar los píxeles.

Observando los resultados obtenidos por el método en conjuntos de datos para el diagnóstico médico se observa que en ninguno de los ejemplos, el número de reglas generadas, supera las 8,6 reglas de media. Siendo la media del número de reglas generadas por el método de 5,22.

Si se observan el número de reglas generadas para el caso en el que se aplica el método a la clasificación en imágenes se concluye que el número de reglas generadas también es bajo, ya que se encuentra en unos valores medios similares a los del método original.

No obstante, el número de reglas generadas, depende en gran parte de la configuración que se indique para los parámetros de entrada del método. Se llega a un punto en el que el número de reglas no varía aunque se permita mediante los parámetros de entrada, pero si que se puede hacer variar ese número entre un mínimo y un máximo que se definen en función del conjunto de datos que se esté estudiando.

4 - BIBLIOGRAFÍA

Artículos

1 – *"An Interpretable Fuzzy Rule-Based Classification Methodology For Medical Diagnosis"* – Gadaras I, Mikhailov L. University of Manchester, School of Computer Science, 2009

2 – *"Towards Interpretable General Type-2 Fuzzy Classifiers"* – Lucas LA, Centeno Tania M, R. Delgado Myriam, ISDA, 2009

3 – *"Conjuntos difusos intervalo-valorados: estado del arte"* – González del Campo R., Garmendia L, Universidad Complutense de Madrid, 2009

PÁGINAS WEB

4 – *http://www.mathworks.com/* – Página Web sobre MatLab, software utilizado para la programación de los métodos.

5 – *http://sitna.navarra.es/* - Página Web del catastro territorial de Navarra, utilizada para la creación del dataset Upna.

6 – *http://campusvirtual.unex.es/* – Página Web de la Universidad de Extremadura con una buena explicación sobre los sistemas basados en reglas difusas.